EMBEDDING SONIFICATIONS IN THINGS

Stephen Barrass
Faculty of Arts and Design,
University of Canberra,
stephen.barrass@canberra.edu.au

Tim Barrass
Independent Artist,
Melbourne, Australia
barrasstim@gmail.com

ABSTRACT

This paper describes three experimental prototypes that explore the embedding of sonifications in things. Through previous work with mobile sonifications we identified requirements for "sonifications in the wild" as being embedded, expendable, and extendeable. Three prototypes, called Flotsam, Jetsam and Lagan, investigate technologies, sounds, materials and metaphors to define and illustrate the design space. The knowledge gained from these prototypes has led to the development of the open source Mozzi sonification library on the Arduino microprocessor.

1. INTRODUCTION

The Sweatsonics mobile phone App was designed for an experiment on sonification in outdoor recreation and fitness activities [1]. During the experiments we found that users often knocked or dropped the phone during these activities, and that breakages are expensive to fix or replace. Ruggedised casings increase robustness and provide waterproofing, but it can be difficult to retrieve a phone, especially if it is in the lake. In reflection on the experiments afterwards we wondered whether it would be possible to integrate the phone into the sporting equipment, rather attaching it to the person who would not usually wear a phone in these circumstances. However touchscreen interfaces are highly visual and difficult to operate if the phone is placed inside another object, and mobile phones are getting larger, not smaller. Finally, these devices are not made to be customised or extended, and the Apple store even restricts the distribution of software. These observations led us to the realisation that it could be more effective to embed sonifications in things designed for the outdoors, rather than in an App on a mobile device designed for the indoors or safe, well known, environments.

Sonifications for the outdoors should be :

- embedded in things
- expendable and cheap to replace
- extendable and customisable

The background section explores alternatives to mobile phones and tablets as a technical platform. The experimental section describes three prototypes that investigate a range of technologies and sensors, forms and materials, metaphors and algorithms. The discussion analyses and reflects on what has been learned from these prototypes. The final section describes the application of the knowledge from these experiments in the development of an embeddable sonification synthesiser implemented on the open source Arduino microprocessor.

2. BACKGROUND

Many domestic products make sounds that convey useful information, and we are all familiar with the beep of a microwave oven and the whistle of a boiling kettle. The CLOSED project at IRCAM explored the design of interactive sonic feedback in products by prototyping a series of abstract Experimental Sonic Objects that synthesised different sonic metaphors in response to user interaction [2]. Different prototypes were shaped to suggest different kinds of interaction, for example the Spinotron had a plunger action that suggested pushing, and the Twister suggested a twisting motion. User interaction was captured by sensors connected to an Arduino embedded inside the object. The sensor data was then wirelessly transmitted to a laptop where the sonic feedback was synthesised using the MAX/MSP software. The sound of the Spinotron was designed to provide continuous feedback about the rate of the pumping action. Pre-trial experiments with a sound acoustically modelled on a ball rolling around a bowl proved to be ambiguous and difficult for users to understand and control. After further iterations and testing the final sound design was based on a model of a fly-wheel ratchet which produced a regular sequence of impact sounds, similar to the freewheel of a bicycle. Pumping the Spinotron caused the fly-wheel to rotate like a spinning top, so that the rate of impacts sped up with the energy of pumping.

The Arduino microprocessor platform provides data acquisition and control functions that allow 8 analog sensor inputs and 8 digital outputs including several pins dedicated to PWM audio output [3]. The system includes the tone() command that can synthesise a sine tone at a specified frequency for a specified duration. The placement of the Arduino inside the Spinotron raises the prospect of embedding the sounds inside the object, rather than playing them on an external laptop. However the variety of sounds that can be synthesised with the tone is limited, especially compared with the MAX/MSP synthesis library. The potential to extend the audio capabilities of the Arduino has been demonstrated by Martin Nawrath who implemented audio buffering and delay lines [4], showing what can be done with a good understanding of the architecture and clever programming. Joe Marshall has also shown what is possible with his Octosynth additive synthesizer that can play sixteen oscillators at once [5]. The Audino is a granular synthesizer for the Arduino that repeatedly plays two triangular waves of adjustable frequency and decay so they merges into a tone that sounds similar to an oscillator with two resonating bandpass filters, producing interesting variations at extreme ranges of the parameters [6]. This algorithm has been deployed by Tobias Grosshauser and Thomas Hermann in an experiment on wearable sonification in which the movement of a dancer was mapped to the frequency and decay of the grains in realtime [7]. The sonification was amplified and sent to a small speaker, and a mobile phone vibrator was used to also provide vibrotactile feedback.

Although these projects have shown the potential to extend the Arduino system to synthesise a wider range of sounds than the built-in tone() permits, the low level programming of the 16MHz integer microprocessor to do audio processing is technically challenging and has limited the development of synthesis algorithms. The needs for students to synthesise interactive sounds in a course on Physical Computing led to the development of the CCRMA Satellite microprocessor system which supports general synthesis software [8]. The Satellite CCRMA board fits into a box of 1300mm x 1750mm x 50mm which is somewhat larger than a smart phone, but still compact enough to be embedded in things such as new musical instruments, dance interfaces, video installations, and guitar effects stompboxes, and the cost of $US125 is comparable to a low end Android phone.

3. EXPERIMENTAL PROTOTYPES

Satellite CCRMA provides a general purpose synthesiser in a compact package that is more embeddable than a phone with a cost at the more expendable end of cheap phones. However, there are Arduino clones that are 10 times smaller costing 10 times less. Although the synthesis capabilities are limited, this level of embeddability and expendability led to the choice of the Arduino for these initial prototypes of embedded sonification. The experiment follows the design-led method of the CLOSED project. The three prototypes, called Flotsam, Jetsam and Lagan, have a nautical theme motivated by our observations of the risk of dropping a mobile phone into the lake in earlier experiments with sonification of rowing. Each prototype combines and explores a mixture of different materials, shapes, technologies, sounds, algorithms, and metaphors with a focus on robustness, waterproofing and floatability in a wet outdoor environment.

3.1. Flotsam

Material floating on the sea, especially debris or goods from ship-wrecks

This first design was inspired by discussions about rowing data where the turning points in graphs of the acceleration of the skiff are used to understand and improve the smoothness and power of the stroke technique. Flotsam, shown in Image 1., is made of wood that floats and protects it from knocks. The two hollowed out halves are screwed together through a rubber gasket to make it watertight.

Image 1: Flotsam prototype.

Flotsam produces impact-like clicks, similar to the Spinotron, that vary in rate with the movement. However the sounds are synthesized inside the object by a Boarduino which is an Arduino clone. The Boarduino was chosen because it has a 75mm x 20mm x 10 mm footprint that fits inside the hand-sized object, because of the relatively expendable cost of $US17, and because it can be inserted into a breadboard for rapid prototyping of external electronic circuits. The sounds synthesized on the embedded Boarduino are transmitted to the outside with a Sparkfun NS73M FM radio chip tuned between 87-108MHz. A 30cm antennae wire wrapped around the inside broadcasts the signal 3-6 metres.
The click is triggered by turning points in any of the 3 axes of acceleration. The rate of clicks increases with the jerkiness of the movement, and stops when the movement is continuous, or stationary. Flotsam could allow a team of rowers to listen in realtime to the turning points in the motion of the skiff through radio headphones, potentially providing a common signal for synchronization. Although the sounds come from the headphones, rather than the object itself, the direct relationship between the clicking and movement creates a sense of causality. The click is synthesised with the built-in tone() at 1kHz for 10ms. In designing sounds on the Arduino it is important to realize that the tone() command blocks all other data acquisition and control functions until it completes. In testing we found that a click of 10ms duration is audible without causing a noticeable interference with the other routines. In trials the FM transmission frequency drifted over time, which required the receiver to be re-tuned after 10-20 minutes, and sometimes it was difficult to find the signal again. However, the biggest problem was interference from commercial radio channels with much stronger signals scattered across the FM spectrum. Unscrewing the halves of the shell to re-tune the transmitter is inconvenient, but an external knob could compromise the robustness and waterproofing. The 9V alkaline battery lasts 10 hours, but unscrewing the halves to replace it is also inconvenient.

3.2. Jetsam

Cargo or equipment thrown overboard to lighten a ship in distress.

This next design iteration was motivated by the observation that it can be difficult to tell someone how to hold a piece of sporting equipment, such as an oar, or a tennis racquet. Could the object itself guide the orientation and position of the user's grip? Jetsam has a twisted shape that can be held in different ways and orientations. This time we investigated haptic vibration as a mode of information feedback that allows the object to be sealed, waterproof and floatable.
Jetsam, shown in Image 2., was carved from pumice which is a natural material formed from lava foam aerated with gas bubbles that can be found floating in the sea or lakes near volcanoes. The exterior was sealed with several layers of polyester to make a smooth and toughened surface. The halves were hollowed out to make space for an Ardweeny clone that has a 40mm x14mm x 10 mm footprint that is half the length of the Boarduino, and which costs just $US10. The 3 axis accelerometer was soldered to the analog inputs on the Ardweeny, which was then inserted in the bottom half, while the 9V battery was inserted into the top half. The two halves are connected by a screw-top (made from a plastic milk bottle) that seals but can be quickly and easily opened to replace the battery and reprogram parameters on the microprocessor.

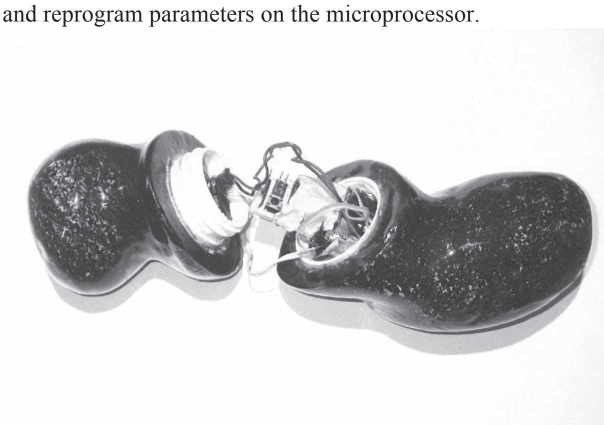

Image 2: Jetsam prototype.

Mobile phone vibrators were glued on the inside at the top end, on one side in the middle, and at the bottom end so that the haptic sensations are localizable. The sense of haptic vibration is subsonic in the range from 0 to 20 Hz after which the feeling becomes continuous. The first experiments with the tone() command to generate haptic effects used frequencies from 0-10Hz but these were not perceptible. This led to the development of a "pulse train" algorithm where each pulse was generated by a 1 kHz tone() with a duration of 50ms which enabled the perception of variation in pulse rate between 0 and 10 Hz. The top vibrator pulses faster in linear steps as the x axis orientation goes from 0 to +-90 degrees. The bottom end pulses in response to y axis orientation, and the middle pulses with the z axis. If the object is held vertical then the top pulses at the fastest rate. A slight horizontal tilt causes the bottom to pulse slowly rate as well. A tilt in all three directions produces pulses at the different rates at the top, bottom and middle. The pattern of haptic sensation changes as the object is held in different orientations. Although the tone() command blocks other routines, the asynchrony of the 3 pulses can be programmed using the millis() function that allows the tracking of time in the control loop. However the scheduling of the control loop varies with CPU load. At orientations with higher pulse rates in all directions the length of the loop changes and the pulsing becomes irregular and glitchy due to the technical limitations of the Arduino to synthesise 3 independent pulse trains at realtime rates. The trials with also showed practical problems with the screwcap attachment. After many openings the wires from the battery became tangled and eventually detached, whilst multiple removals of the Ardweeny from the cavity detached the solder connections to the vibrators.
This prototype also raises the question of how accurately a user could learn to estimate the orientation of the object from the 3 localised pulses.

3.3. Lagan

Cargo or equipment thrown into the sea but attached to a float or buoy so that it can be recovered.

The third iteration, called Lagan, further explores issues identified in the Flotsam and Jetsam experiments. In this version we explored the effect of producing the sound directly from the object, investigated an extension of the system to create a customized low level synthesis algorithm as an alternative to the tone() command, and looked into ways to technically workaround the limits of audio blocking and CPU load. Lagan is crafted from a cuttlefish backbone, which is another natural material that floats and provides shockproofing. Two half shells were hollowed out and laquered with polyester to seal them. The halves were screwed together with a gasket, due to the problems with the screwtop connector in Jetsam. This prototype used the Arduino Nano clone with a 43x19x10 mm footprint similar to the Ardweeny, but with a lower voltage that allows a reduction in size with a flat 6V Li-ion battery. Stereo speakers were made by glueing flat piezo buzzers against holes drilled through the shell that were covered with thin plastic to make water resistant speakers directly on the object.

Image 3: Lagan prototype.

The sonification in Lagan was designed to have a sea wave-like sound to match the organic shape and material. Because the tone() command can only produce beeps we needed to find another way to synthesise these sounds. Drawing on the Audino granular synthesizer, the sound was synthesised by varying the amplitude of white noise loaded into a cycling buffer and output to the PWM analog audio output. The need to synthesise a continuous sound without blocking data acquisition and control required the default timer behaviour to be overridden. The continuous variation in the amplitude of the noise sonifies the continuous variation in the acceleration in all directions. The sound is audible but an amplifier could improve the dynamic range.

Flotsam, Jetsam and Lagan were shown as an installation curated for the Conference on New Interfaces for Musical Expression in Sydney in 2010 [9], as shown in Image 4.

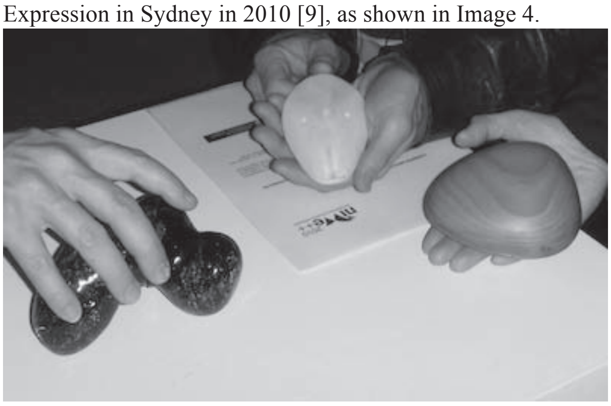

Image 4: Flotsam, Jetsam and Lagan installation at NIME 2010.

4. DISCUSSION

Flotsam demonstrated that useful sonifications can be designed with the tone() command on an Arduino that is small and cheap enough to be embedded in things used in demanding outdoor activities. It raised issues with the range of sounds that can be produced, and the technical problems of transmitting sounds from inside the object to the outside. This raises questions about the schizophonic separation of the sound from the object in sonification?

Jetsam demonstrated that haptic vibrations could also be a mode for information in things. Are haptic audio vibrational representations of data a subset of sonification or another field entirely e.g. vibrification? This experiment also reiterated technical issues around the blocking behaviour of the tone() command that impact multichannel sonic and haptic feedback on the Arduino.

The development of a continuous custom designed sound for the Lagan prototype required changes to the low level workings of the Arduino software. This development adds a new capability to synthesise continuous sonic feedback without blocking the data aquisition and control. Building on this technical breakthrough we developed a continuous noise synthesizer that can be interactively varied in amplitude in response to a continuous stream of realtime data from a sensor.

5. CONCLUSION

These experiments have provided insights and understanding of the design space of embedded sonifications. We found that the Arduino microprocessor can be an alternative to consumer mobile devices as a platform for sonification in outdoor activities. However the *tone()* command limits the range of sounds to beeps and sine tones, and the blocking behaviour affects the interactivity and scheduling, hindering the the development of more than very simple applications. However we were able to take over the timers on the Arduino to program a custom synthesis algorithm that does not block, opening up the potential to design much more complex sonifications. The capability to extend the functionality of shows the power of open source software to facilitate research and development. These experiments have led to the initiation of an open source sonification synthesizer project called Mozzi on github [10]. You can download and freely use Mozzi under the open source license in your own embedded sonifications [10]. You can also add your own extensions by forking the source. We hope that Mozzi will enable a wave of embedded sonifications.

6. REFERENCES

[1] Barrass, S., Schaffert, N., Barrass, T. (2010 Probing Preferences between Six Interactive Sonifications Designed for Recreational Sports, Health and Fitness, in Proceedings of ISon 2010, 3rd Interactive Sonification Workshop, KTH, Stockholm, Sweden, April 7, 2010.

[2] Lemaitre, G. Houix, O. Visell, Y. Franinovic, K. Misdariis, N. and Susini. P. (2009) Toward the design and evaluation of continuous sound in tangible interfaces: The Spinotron. International Journal of Human-Computer Studies, Volume 67 Issue 11, November, 2009, Pages 976-993, Academic Press, Inc. Duluth, MN, USA.

[3] Arduino, retrieved 31 March 2013, http://www.arduino.cc/

[4] Nawrath, M. Arduino Realtime Audio Processing, retrieved 31 March 2013, http://interface.khm.de/index.php/lab/experiments/arduino-realtime-audio-processing

[5] Marshall, J. Arduino Octo-Synth, retrieved 31 March 2013, http://www.cs.nott.ac.uk/~jqm/?p=605

[6] Audino synthesiser, retrieved 31 March 2013, http://code.google.com/p/tinkerit/wiki/Auduino

[7] Grosshauser T. and Hermann T. (2010), Wearable Setup for Gesture And Motion Based Closed Loop Audio-Haptic Interaction, ICAD 2010: Proceedings of the 16th International Community for Auditory Display, Washington, USA.

[8] Berdahl E. and Ju W. (2011) Satellite CCRMA: A Musical Interaction and Sound Synthesis Platform, Proceedings of the International Conference on New Interfaces for Musical Expression, pp. 173–178, 2011.

[9] Barrass S. and Barrass T. (2010) Flotsam, Jetsam and Lagan, in Muller L. (curator) Room Based Installations, Proceedings of the Conference on New Interfaces for Musical Expression, Sydney, 2010, http://www.educ.dab.uts.edu.au/nime/program.php#Installation

[10] Barrass, T. Mozzi Sound Synthesis Library for Arduino, retrieved 31 March 2013, http://sensorium.github.com/Mozzi/